Visualizing the Passage of Time with Video Temporal Pyramids

Melissa E. Swift, *Member, IEEE*, Wyatt Ayers, Sophie Pallanck, and Scott Wehrwein

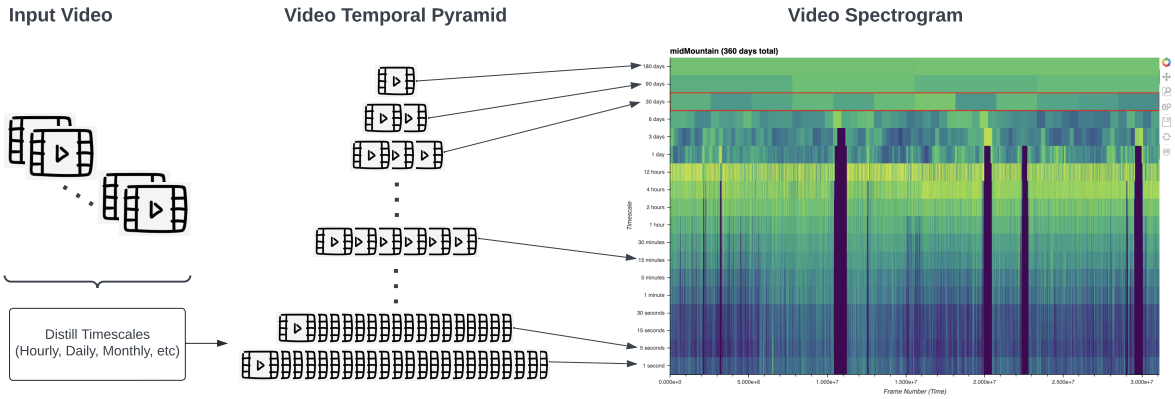


Fig. 1. Given months or years of recorded webcam footage, our approach builds a Video Temporal Pyramid consisting of different length shorter videos, each of which visualizes the events happening at a particular timescale. Our Video Spectrogram is a visualization for the pyramid that provides both overview and drill down functionality to aid in interactive exploration.

Abstract— What can we learn about a scene by watching it for months or years? A video recorded over a long timespan will depict interesting phenomena at multiple timescales, but identifying and viewing them presents a challenge. The video is too long to watch in full, and some things are too slow to experience in real-time, such as glacial retreat or the gradual shift from summer to fall. Timelapse videography is a common approach to summarizing long videos and visualizing slow timescales. However, a timelapse is limited to a single chosen temporal frequency, and often appears flickery due to aliasing. Also, the length of the timelapse video is directly tied to its temporal resolution, which necessitates tradeoffs between those two facets. In this paper, we propose Video Temporal Pyramids, a technique that addresses these limitations and expands the possibilities for visualizing the passage of time. Inspired by spatial image pyramids from computer vision, we developed an algorithm that builds video pyramids in the temporal domain. Each level of a Video Temporal Pyramid visualizes a different timescale; for instance, videos from the monthly timescale are usually good for visualizing seasonal changes, while videos from the one-minute timescale are best for visualizing sunrise or the movement of clouds across the sky. To help explore the different pyramid levels, we also propose a Video Spectrogram to visualize the amount of activity across the entire pyramid, providing a holistic overview of the scene dynamics and the ability to explore and discover phenomena across time and timescales. To demonstrate our approach, we have built Video Temporal Pyramids from ten outdoor scenes, each containing months or years of data. We compare Video Temporal Pyramid layers to naive timelapse and find that our pyramids enable alias-free viewing of longer-term changes. We also demonstrate that the Video Spectrogram facilitates exploration and discovery of phenomena across pyramid levels, by enabling both overview and detail-focused perspectives.

Index Terms—Time, time-frequency, video visualization, multi-scale, webcam

1 Introduction

The world around us is constantly changing at many speeds at once, but the human visual system can only perceive a narrow range of dynamic phenomena in real time. Some things move too slowly for us to register, such as a glacier flowing, and some things move too quickly for us to register, such as a bee's wings in flight. Though we cannot see these motions as they occur, we can visualize them after the fact. For fast events, we can use a high-speed camera and slow down the footage to a more human-friendly speed (i.e., slow-motion). For slow events, the most common method of visualizing these changes is timelapse photography, where frames are taken at a regular intervals over time and then assembled into a video that plays much faster. The key observation that motivates our work is that many scenes exhibit interesting phenomena at multiple timescales: in a single scene, we might be able to observe foot and vehicle traffic on a road, movement of clouds in the sky, diurnal changes in illumination, and a building being constructed over the course of months or years.

A natural way to capture these multi-timescale phenomena is to begin with an input video with sufficiently high framerate to capture the fastest-moving phenomena. However, months of raw video cannot be viewed in a reasonable amount of time, so we might subsample it to create a series of timelapse videos that show different rates (e.g., one frame per minute, one frame per hour, etc.). However, straightforward timelapse sampling exhibits aliasing due to high-frequency content. Consider sampling one frame per month; although longer-term changes happening at or around a months-long timescale will be natu-

- *Melissa E. Swift conducted this research while at Western Washington University and is currently with Pacific Northwest National Laboratory. E-mail: melissa.swift@pnnl.gov.*
- *Wyatt Ayers is at Western Washington University. E-mail: ayersw2@wwu.edu.*
- *Sophie Pallanck was at Western Washington University. E-mail: sophierosepallanck@gmail.com.*
- *Scott Wehrwein is at Western Washington University. E-mail: scott.wehrwein@wwu.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

rally viewable, shorter-term changes, such as a person that happened to be walking through the scene at the moment a frame was sampled, will appear as a distracting single-frame blip. This paper proposes (1) Video Temporal Pyramids, a more principled, alias-free approach for visualizing changes at different timescales; and (2) the Video Spectrogram, a visualization tool for navigating and exploring the pyramids.

The algorithm that forms the basis for creating a Video Temporal Pyramid takes inspiration from the common image processing techniques of Gaussian and Laplacian image pyramids, but applied to the temporal domain. The result is a collection of new videos, each of which distills the changes happening at a particular timescale (e.g., hourly, daily, monthly, yearly). Though each pyramid level is similar to a timelapse with a particular sampling rate, they feature a smoother viewing experience with no aliasing or flickering effects.

The Video Temporal Pyramid captures information about the changes over many timescales, but the volume of video data is (approximately $2\times$) larger than the original. To help a user navigate and explore the pyramid and surface more information about events and patterns in the scene, we propose a visualization tool called the Video Spectrogram. We quantify the magnitude of changes happening at each time and in each timescale and plot those data as a heatmap, analogous to the spectrogram used in audio processing [10]. The resulting spectrogram plots time vs. timescale, showing clear patterns for strong cyclic changes such as day/night and seasonal periodicity. Anomalies such as significant weather events and corrupted data can also be discovered. The Video Spectrogram facilitates connecting the video footage to specific dates and times when events occurred. This enables the user to quickly do an overview scan and then drill down to lower timescales to view more details for a particular day or time, in keeping with Shneiderman's information seeking mantra [37]. This is key to making the large volume of video data manageable without arbitrarily discarding information.

To validate our contributions we have processed multiple long-duration webcam datasets of diverse outdoor scenes, including a construction site, a ski slope, and a mountain lake, among others. The time periods covered by our datasets range from 1 month to 16 years, with base temporal resolutions ranging from 30 frames per second to 1 frame per hour (Fig. 6). In our exploration of these datasets, especially in direct comparison to a timelapse baseline, we found that our pyramid videos and spectrogram tool allowed us to rapidly learn a lot of detailed information about each scene, from how the seasons changed all the way down to the exact time a particular object appeared in the scene. Please refer to our supplementary materials to view selected Video Temporal Pyramid videos from each of our datasets as well as a demo of the Video Spectrogram.

2 Background and Related Work

The proposed Video Temporal Pyramid and Video Spectrogram are closely related to work in several subdisciplines. This section provides a brief overview of the most relevant.

2.1 Timelapse and Related Techniques

Timelapse has been around since the late 1800's [46] as a popular way to visualize the passage of time on a small scale (*e.g.*, a pineapple rotting [39]) or a large scale (e.g., Google Earth Timelapse [12]). Of particular interest, Martin-Brualla *et al*. [25] drew from internet imagery to create years-long timelapse videos, but the noise from internet photos captured by different cameras and at different times requires heavy smoothing so that shorter-term changes are not visible and long-term changes can be hard to detect.

Several techniques have been proposed to smooth out the aliasing artifacts that result from timelapse sampling after the fact. With consumer video applications (*e.g.*, hyperlapse and timelapse) in mind, Zhang *et al*. [49] propose a method for smoothing transitions between frames in temporally subsampled videos. Whereas their method interpolates and smooths after subsampling, our method necessitates fewer modeling assumptions because we smooth discontinuities before temporal subsampling; starting with the densely sampled input also allows us to produce smooth visualizations of any timescale. Their method is also tested only on videos that span minutes or hours of time, and are subsampled to seconds or minutes, whereas ours is designed to work with years-long video streams. Finally, their method is also more computationally expensive, operating around 0.5 frames per second, whereas ours runs at 6 frames per second. While both methods can benefit from parallelization, this is nonetheless a significant difference when considering datasets such as ours that have on the order of 1 billion frames. Details on runtime and datasets can be found in the supplemental material.

In remote sensing, it is often desirable to visualize long-term changes related to a variety of phenomena caused by humans or natural events. The data is often of low temporal frequency, sometimes only before-and-after satellite pictures or landscape photographs taken far apart in time, such as the U.S. Geological Survey repeat photography project [41]. Animation techniques have been proposed which can help create a smooth transition between images. Lobo *et al*. [24] does this by simulating plausible intermediary frames, while Harrower [15] provides the user with control over the spatial and temporal resolution to allow for optimal visualization of a given phenomenon. While our work shares the same goals of visualizing changes happening over long timescales, we work with datasets with high temporal resolution and do not rely on interpolation to smooth transitions. That said, interpolation methods such as Lobo *et al*. [24] could be complementary to ours as a possible way to fill in segments of missing data from our datasets.

2.2 Temporal Resampling and Video Visualization

Most existing techniques for video visualization are designed for videos no longer than a few hours, and their end goals often differ significantly from ours. In fact, over ten years ago, Borgo *et al*. [5] published a survey of different video visualization techniques; while our work is related to many of the techniques they describe, the authors clearly assumed the use of relatively short videos. This section highlights some of the most closely related work in this area.

Various works adapt the frame rate, or temporal sampling rate, of a video over time based on its content [19, 51]. The most closely related technique is Computational Timelapse [4], which uses temporal differences in video frames to dynamically speed up the frame rate when little is changing and slow it down when more changes are occurring. While this is effective as an automatic fast-forward tool, it requires a chosen output video length; furthermore, long-term changes appear choppy and broken up due to sudden changes in the frame rate. Several works have proposed various non-axis-aligned manipulations of space-time cubes such as videos; Rav-Acha *et al*. [30] explore this idea from in a graphics/vision context, while Bach *et al*.give a thorough visualization-oriented review of the possibilities in this space. These techniques are generally incompatible with cuboids with significantly longer time extent, such as months-long videos.

Video summarization techniques take another approach—rather than maintaining chronological and spatial continuity, these techniques attempt to find frames or clips that encapsulate the activity in a video. These techniques generally approach the problem by automatically detecting "noteworthy" frames or clips either by using unsupervised saliency-based approaches [29] or by using example-based learning [31, 48]. These techniques tend to focus on the real-time timescale and treat longer-term changes as noise; they also make automated decisions that may remove content of interest even at the real-time timescale.

Video fast-forward techniques are closely related to the ideas supporting timelapse videos and often use frame-skipping. The disadvantages of timelapse were explored by Hoferlin *et al*. [17]. Their evaluation of fast-forward techniques also included an interesting experiment with the use of temporal blending—similar in spirit to our temporal filtering method. However, their filtering approach does not generalize to more than one timescale.

A few prior works have considered multiple timescales of activity in videos. Motion Denoising [34] separates a video into short-term and long-term components. This technique produces excellent results, but handles only two timescales and is very computationally expensive, making it impractical for months-long videos, much less for years-long videos. Wehrwein *et al*. [45] propose a method to composite clips

from different timescales into a single "scene summary", showing, for example, people walking, clouds moving, and the sun crossing the sky all at once. This method begins with a Gaussian temporal pyramid much like the one constructed as a side-effect of our Laplacian pyramid construction, then composites salient clips from different pyramid layers together; though multiple timescales are visualized at once, the vast majority of the lower pyramid levels are discarded from the final output. In contrast, we construct visualizations that assists in exploration of the whole dataset without any assumptions about which timescales or scene elements are of interest to the viewer.

2.3 Interactive Video Exploration and Retrieval

In contrast to the automated methods above, a separate category of prior work facilitates interactive browsing, exploration, and retrieval in video. Though this is more closely related to our task, most of these techniques are oriented towards retrieval of specific content rather than discovery, or towards improving the ability to scrub or seek in a shorter video.

The Video Browser Showdown [33] is a yearly contest of video browsing tools designed to locate either a specific event or clip in a video, or locate all instances of an event or action. Tools that are successful on this task (e.g., [21]) tend to leverage the fact that the sequence of interest is known *a priori*, making them less useful for discovering unknown anomalies; for similar reasons, these tools are also unlikely to generalize well for the purpose of identifying structure at longer timescales.

Several techniques have been proposed to show a visual overview of an entire video sequence, or improve scrubbing. Gutwin *et al*. [13] proposed a spread-loading scheme to load frames at varying intervals when loading a streaming video to improve the scrubbing experience. They showed that this improved users' ability to seek to a particular point in the video quickly, but even with instantaneous availability of all frames, a months-long video would be tedious to explore by scrubbing. Barnes *et al*. [3] create a continuous visual overview of automatically selected keyframes to assist in scrubbing around in the video, while Jackson *et al*. [18] arrange short clips of a video in an animated grid so a user can shift their focus to any point in a video or watch a single thumbnail as it cycles through the entire video. These techniques work well for shorter clips, but their utility is limited by available screen size for longer-duration videos.

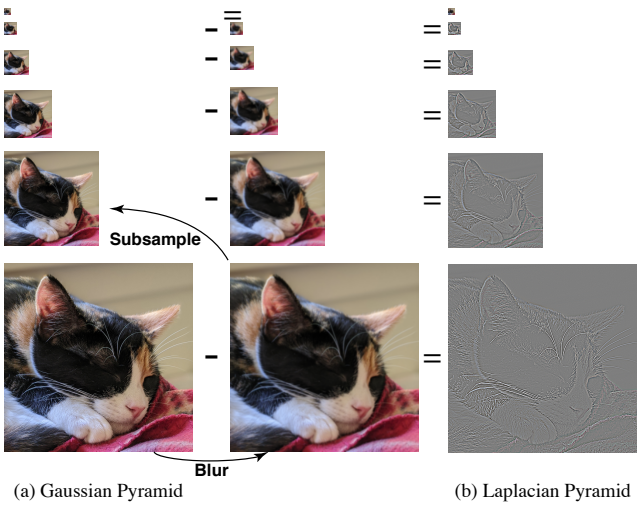
From a visualization perspective, Romero *et al*. [32] also uses computer vision to analyze and visualize video volumes. In particular, their interface proposed a closely related heat map visualization called the Activity Table that is similar in spirit to our Video Spectrogram. The Activity Table displays aggregate motion computed using thresholded frame differences, closely related to the frame differences performed in the construction of our Laplacian Pyramids. Whereas they plot aggregate motion in a single (real-time) timescale across different spatial locations, we are interested in activity at multiple timescales and use the vertical axis of the heatmap to index temporal frequency instead of spatial location.

2.4 Time Series Visualization

Although our work relates to the rich literature on time series visualization (e.g., [1, 2]), video has unique properties that benefit from domain-specific techniques. One notable example from this literature is the work by Cakmak *et al*. [9] which does visualize time-varying data at multiple timescales; their interface for traversing temporal scales and viewing summaries at different levels is loosely analogous to our Video Spectrogram, but is geared towards the very different domain of time-varying graph data.

2.5 Pyramids and Spectrograms

The pyramid computation component of our approach is directly adapted from image pyramid techniques from the computer vision literature. Our adaptation will be described in detail below. In particular, we compute Gaussian [8] and Laplacian [7] pyramids along the time dimension, in contrast with their traditional application to the spatial dimensions of images. The idea of generalizing image pyramid techniques to videos is not new; a related generalization of the Gaussian pyramid to video was proposed by [11] to send videos at variable resolutions in space and time over limited bandwidth network connections. Their approach resembles a Gaussian pyramid (in contrast to our Laplacian pyramid), operates across spatial and temporal dimensions, and is designed for efficient coding and variable-resolution transmission of videos under limited bandwidth. Our method aims to visualize longer-term changes with high fidelity. For the problem of human action recognition, numerous works propose variations of pyramids applied temporally for short (e.g., minutes-long) video clips [22, 36, 43, 44, 50]. To our knowledge, however, our method is the first to extend the temporal Laplacian pyramid concept to extremely long-duration videos with the intent to visualize long timescales. Finally, our Video Spectrogram tool is directly inspired by the idea of time-frequency representations like the spectrogram, which are commonly used in audio visualization and processing [10].



(a) Gaussian Pyramid (b) Laplacian Pyramid

Fig. 2. Traditionally, Gaussian and Laplacian pyramids are applied to both spatial dimensions of an image. The left column (a) shows a Gaussian pyramid each layer of which is blurred and subsampled from the prior one. Each level of the Laplacian pyramid (b) is a high-pass filtered version of the corresponding Gaussian level, computed by subtracting the blurred level from the original.

3 Video Temporal Pyramids

Our approach is inspired by image pyramids from the computer vision literature, which allow for separation and manipulation of spatial frequencies in images. We generalize these same techniques to the temporal domain in videos in order to separate and manipulate temporal frequencies. Specifically, the core of our Video Temporal Pyramid approach is a Laplacian pyramid computed in the time dimension, approximating the output of a bank of band-pass filters applied pixel-wise across time. We first formally define the pyramid's construction, then discuss several adaptations for the domain of long, fixed-camera videos.

3.1 Definition and Construction

Image pyramids are a classical technique from the computer vision literature [7, 8], widely used to apply image processing and computer vision algorithms at multiple scales or in a scale-invariant fashion. The most basic image pyramid is a Gaussian Pyramid [8] (Fig. 2 (a)), constructed by repeatedly blurring then subsampling an image. Each subsequent level of the pyramid represents what remains after a low-pass filter is applied to the prior level. Each level of a Laplacian pyramid [7] (Fig. 2 (b)) represents the result of a *high-pass* filter applied to the prior level of the Gaussian pyramid, or equivalently a *band-pass* filter applied to the original image. The resulting Laplacian pyramid levels contain narrow slices of spatial frequency content of the image,

thereby resembling the output of a bank of band-pass filters.

Where Laplacian pyramids are traditionally used to isolate and manipulate *spatial* frequency content of images, we instead apply the same procedure to the *temporal* dimension of a video, leaving the spatial dimensions alone. A temporal analog to the Gaussian pyramid consists of videos that have been filtered and subsampled along the time dimension only. The Laplacian pyramid is also constructed analogously, by subtracting the temporally blurred video from the original. In principle, frames of the Laplacian temporal pyramid can be computed by subtracting the computed blur frames from the current level of the Gaussian pyramid before subsampling. In practice, we use the standard pyramid construction approach given by [7] to avoid quantization errors and ensure that the pyramid levels can accurately reconstruct the input signal. We first filter and downsample the input, then upsample it again to match the prior level's temporal sampling rate; this downsampled-then-upsampled signal is finally subtracted from the input video to calculate the Laplacian pyramid level.

The pyramid levels are constructed recursively as shown in Fig. 3 and Algorithm 1. We begin with a long input video, which serves as the first level of the Gaussian temporal pyramid. Each subsequent level is computed in one pass through the prior level's video, calculating blurred frames from a sliding window of prior level frames. The resulting pyramid levels are themselves videos of the same spatial resolution and covering the same real-world duration in time, but with a reduced frame count. For this reason, each level of a Gaussian temporal pyramid is similar to a timelapse video with a particular sampling rate. One key difference is that blurring across time before subsampling causes short-term motions to blur out in higher levels of the pyramid, thus eliminating aliased content that would appear in a true timelapse video. The levels of the Laplacian temporal pyramid are less intuitive to watch, as they contain only specific bands of temporal frequency content. Sample Laplacian pyramid frames are shown in Fig. 4.

Algorithm 1 Construct temporal pyramid

Input: V, an input video of size $(H \times W \times C \times$ Frames$)$
Output: $G = G_{1\ldots N}$, the Gaussian pyramid
Output: $L = L_{1\ldots N}$, the Laplacian pyramid
for $i \leftarrow 1 \ldots N$ **do**
  $F \leftarrow filter(V)$ ▷ apply linear 1D blur in time
  $V' \leftarrow subsample(F)$ ▷ *e.g.*, if stride=3, keep every 3rd frame
  $G_i \leftarrow V'$
  $F^* \leftarrow filter(upsample(V'))$ ▷ account for quantization error
  $L_i \leftarrow V - F^*$
  $V \leftarrow V'$ ▷ set up input for next level
end for

In a Laplacian temporal pyramid the original source video has been decomposed into multiple component videos. These components can be reassembled to create an exact copy of the source by performing the pyramid-creation steps in reverse order (see Algorithm 2). It is also possible to proceed with this reconstruction while leaving out certain Laplacian pyramid levels (specified in Algorithm 2 using the W vector). Reconstructing without the last few Laplacian layers yields a smooth but slower-moving version a Gaussian blur level. We found this smooth temporal upsampling option to be quite useful for some of our videos, as it provides a way to slow down the action so more information can be absorbed by the viewer.

Each level of the pyramid represents a specific *timescale*. For instance, the base frame rate of most webcam video is 30 frames per second (fps), or 1/30th of a second per frame. Motions easily visible at this frame rate can be thought of as belonging in the "1/30th of a second" timescale. Higher levels of the pyramid represent changes happening at a frequency of "once per second" or "once per day" or even "once per month."

3.2 Adaptations for Months of Static-Camera Video

The prior section described a straightforward generalization of the Gaussian and Laplacian image pyramids to construction of temporal pyramids. We now describe a few simple adjustments we made to adapt these pyramids for the use case of visualizing and exploring long, fixed-camera video streams.

Algorithm 2 Reconstruct pyramid level videos or upsample

Input: $k \in \{0 \ldots N-1\}$, chosen ending level
Input: $L_{k+1\ldots N}$, Laplacian temporal pyramid videos
Input: G_N, the top level Gaussian temporal pyramid video
Input: $W \in \{0,1\}^N$, indicator vector of detail levels to reconstruct
Output: R_k, reconstructed video from level k
$B \leftarrow G_N$
for $i \leftarrow N \ldots k+1$ **do**
  $U \leftarrow upsample(B)$ ▷ *e.g.*, if stride=3, repeat each frame 3 times
  $F \leftarrow filter(U)$ ▷ use same filter from pyramid construction
  $B \leftarrow F + (L_i \times W_i)$ ▷ add detail layer if applicable
end for
$R_k \leftarrow B$ ▷ reconstructs original when $k = 0$ and $W_{k+1\ldots N} = 1$

3.2.1 Variable Downsampling Rates

In image pyramids, the filter width and subsampling rate are parameters that are traditionally tuned according to the application. For example, to achieve scale invariance for computer vision algorithms, a subsampling rate smaller than 2 is often desirable to detect objects at a densely sampled range of sizes. In our application, the pyramid is unlikely to miss anything, even with a larger sampling rate because motions and dynamics tend to be visible in a range of timescales. For example, in a 30 frames-per-second (30fps) video of a person walking through a courtyard, the person might take 6 seconds to walk through the scene, and thus would appear, moving increasingly quickly and increasingly blurred, in at least the first four or five levels of the pyramid.

While we initially simply used a downsampling rate of 2, problems arise when the temporal sampling rate of each pyramid level is not aligned with intuitive units of time. As discussed in [1], modeling time has many complexities to consider. For example, if our input video (Gaussian pyramid level 0) is captured at 1/30 second per frame (30 frames per second) and we chose a fixed scale factor of 2x, then level 5 of the pyramid would cover 1.066 seconds per frame, level 10 would cover 34.133 seconds per frame, and level 22 would cover 1.618 days per frame. In addition to being less intuitive for interpretation, these sampling rates can introduce aliasing at higher sampling rates due to periodic phenomena such as the day/night cycle or seasonal changes. Since our goal is to visualize and discover structure at long time-scales, it is important to have sampling rates lined up with known patterns such the 24-hour cycle of the day and the 365-day cycle of the year. Achieving this alignment requires choosing different subsampling rates for different levels of the pyramid. See Supplementary Material for details of the sampling rates used and timescales represented at all levels of our pyramids. We used strides of 2, 3, and 5, which necessitated the use of different blur filters depending on stride. The one-dimensional blur filter applied across frames was [1,2,2,1] when the stride was 2; [1,2,3,2,1] when the stride was 3; and [1,2,3,4,5,4,3,2,1] when the stride was 5.

3.2.2 Scaling to Months and Years of Video

The algorithm as described thus far requires a full pass through each pyramid level to compute the next; because the layers sizes shrink exponentially, this requires the equivalent of roughly 2 passes through the full input video, for an $O(n)$ runtime. However, for months-long input videos such as ours, this is still very slow and can be easily parallelized. To accelerate the computation of pyramids, we compute one-day pyramids in parallel on a cluster, then merge the one-day pyramids to compute the higher pyramid levels. These one-day pyramids are computed up through level 15, where the Gaussian blur video for the entire day is 1 frame, and the corresponding Laplacian pyramid level shows activity changing on a 12-hour timescale. The one-day pyramids are then merged by stitching together each day's 1-frame

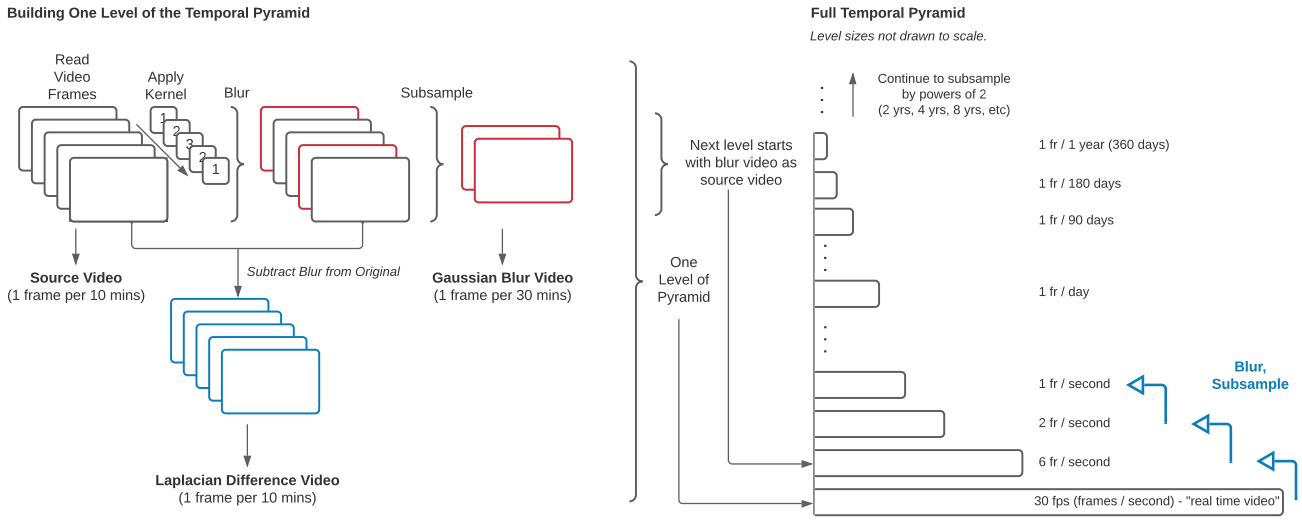


Fig. 3. An overview of the temporal pyramid construction process. **Left:** an illustration of the algorithm for computing one level of the Gaussian and Laplacian pyramid. A source video (the input, or a prior level of the Gaussian pyramid) is blurred in the temporal dimension. The subsequent level of the Gaussian pyramid is computed by subsampling this blurred video, while the Laplacian pyramid level is computed by subtracting the blurred video from the source. **Right:** The resulting pyramids are the collection of videos generated using the above algorithm.

Gaussian 'video' into a full blur video for level 15, which is then used as the source video for the construction of the remaining pyramid levels.

We also parallelize the creation of years-long videos in the same manner, running each year separately and then stitching them together and continuing the construction of multi-year pyramid levels. In addition to being efficient, this also allows us to approximate a 365- (or 366-) day year with only 360 frames, which is necessary in order to use only sub-sampling rates of 2, 3, and 5. For each individual year, after analyzing which days have the most missing frames, we choose 5 days to remove from the pyramid (or 6 days for a leap year). If we used this 360-day year and did not compute each year separately, we would end up with a true 360-day timescale and some aliasing over the course of multiple years, where the year shown would slowly get out of alignment with the calendar year. When we parallelize the years, we end up with 1 frame per year at level 21 (the 1-year timescale), which line up with calendar years, and the higher levels can be built on that solid foundation. We currently sub-sample with powers of 2 for multiple-year timescales, but it would be possible to sample by 2 and then 5 (or 5 and then 2) in order to create a 1-decade timescale.

Missing data is filled in with all-black frames in our temporal pyramid videos. See the supplementary material for a more detailed description of how we handled missing data.

4 Video Spectrograms

At this point, we have described an approach for parsing out temporal frequency content of a long video stream into timescales by constructing a temporal Laplacian pyramid. Watching just the upper level videos of the pyramid is an efficient way to gain an overview of temporal dynamics and long-term events because they are short but distill important information. However, the pyramid itself does not make it any more tractable to watch the entirety of the lower levels, which still have very long durations. To help address this, we propose a visualization tool called the Video Spectrogram that facilitates interactive navigation and exploration of the pyramid levels.

The Video Spectrogram user interface evolved to include multiple elements, as shown in Fig. 5; however, the main element and key idea is a 2-dimensional plot that provides an overview of the entire pyramid by showing time on the horizontal axis and timescale (frequency) on the vertical axis. Each cell in this time/frequency grid represents a 2D frame from one of the pyramid videos, which would be unwieldy to visualize in such a small space; instead, we abstract the spatial details and display a single quantity that captures aggregate activity.

By construction, the Laplacian pyramid layers are "difference" frames representing only content that has changed at the corresponding timescale. Therefore, the aggregate activity for a given frame in a timescale can be measured by taking a norm of the Laplacian frame. We chose the L^2 norm (i.e., the square root of the sum of squared pixel values), and display it on a logarithmic scale. We experimented with other norms (L^1) and color scales (linear). Because we aggregate across pixels, the L^2 norm gives more weight to spatially smaller changes with larger magnitude versus more widespread, smaller-magnitude changes. The logarithmic color map does a better job of showing contrast in low-activity regions, allowing subtler patterns to be detected when overall activity levels are low.

We compute the norm for each frame in each Laplacian pyramid level, and display the resulting values as a 2-dimensional heatmap as shown in Fig. 1 and Fig. 5, where each tile in the heatmap is the norm of one Laplacian pyramid frame. Tiles in higher timescales become wider because the same temporal extent is represented using fewer frames at higher pyramid levels.

The temporal pyramid videos and the spectrogram plot are closely linked; the purpose of the spectrogram is to help explore the pyramid, so we include a large and prominent video player to show the pyramid videos. The full-spectrogram plot is good for an overview; however, we found that since we generally watch the video from one level at a time, it was useful to enlarge the portion of the heatmap corresponding to the level being watched in the video. We visualize this single-level spectrogram below the video, along with a moving vertical line that travels along the plot as the video plays. As the user sees an event unfold in the video they can get a sense of what the spectrogram shows during the event. The full-spectrogram plot always includes a red outline marking the level and/or date being viewed in the current video, for a 'you are here' connection to the bigger picture. Both plots have pan and zoom functionality to assist with overview-to-detail visualization.

In order to go in the other direction, and see what the video content is like at a particular spot by starting from the spectrogram, we implemented a mouse-over functionality whereby a thumbnail image of the corresponding video frame shows up underneath the plot when the mouse hovers over a cell of the single-level spectrogram. Easy access to those thumbnail images gives the user hints about the reason for the

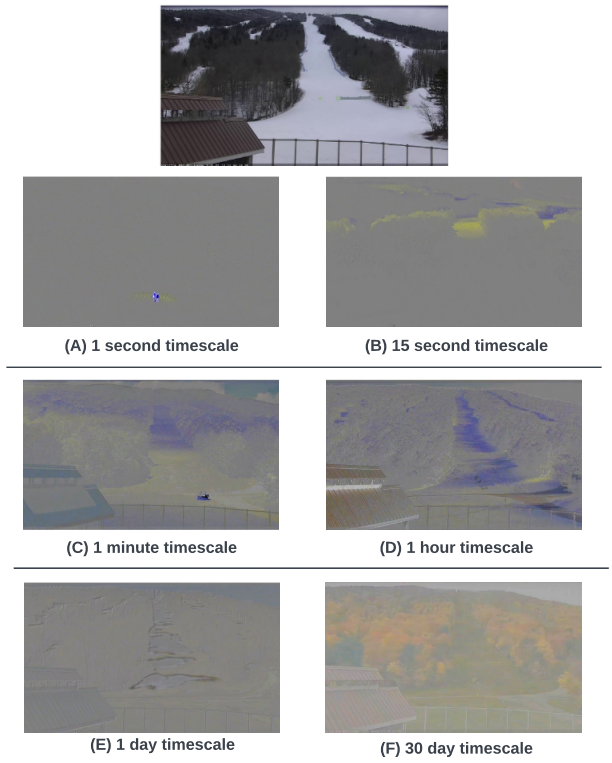


Fig. 4. Examples of frames from Laplacian Difference videos at different frequencies, taken from the same webcam but not necessarily from the same days. These show the pixels that changed during that time span. (A) people walking; (B) evidence that the sun peeked out from behind some clouds; (C) a golf cart or similar slow-moving vehicle; (D) evidence of the sun's movement across the sky which has cast shadows of the trees on the ground; (E) outlines of snow patches which likely means that those patches melted over the course of the day; (F) most elements of the scene are visible, including fall colors, which likely means that the autumnal seasonal changes that month affected most pixel values.

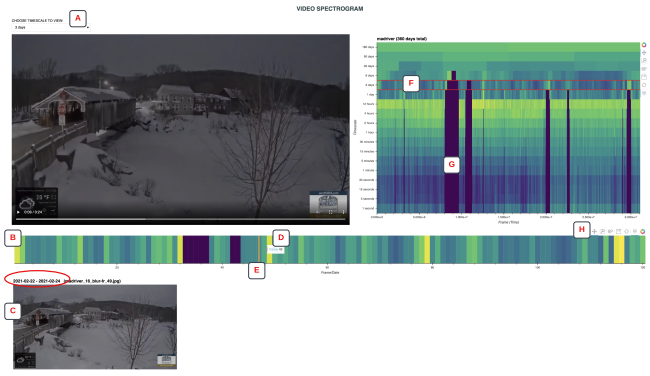


Fig. 5. Drop-down menu (A) for choosing level to view. Video player updates with appropriate level video. Chosen level is highlighted on the full spectrogram (F) as well as enlarged below the video (B). On mouse hover (D) a thumbnail image of that frame shows below (C). As the video plays, a vertical orange line (E) will travel along the single-level spectrogram plot, aligning the date/time between video and plot. Areas with missing data, such as (G) show up clearly. Toolbars (H) allow for zoom and save.

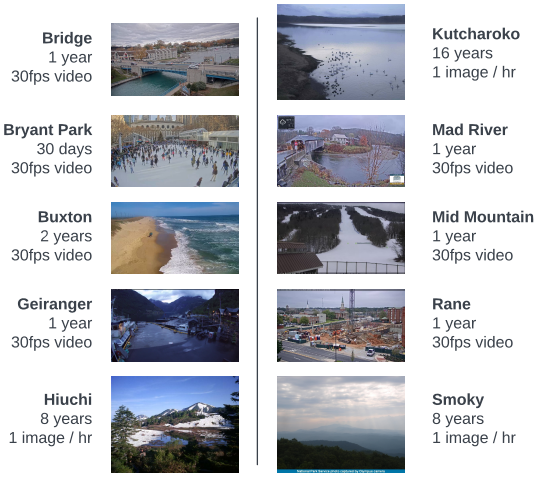


Fig. 6. Sample frames from our datasets, along with the names we are using to refer to them in this paper, their covered timespan, and their base frame rate. More details in supplemental material.

structure in the heat map and helps determine whether it's useful to drill down or investigate further in that area. The thumbnail-on-hover functionality also makes it possible to "scrub" through the video for that timescale by dragging the mouse horizontally over the plot at any speed.

Users can navigate to different levels of the pyramid using a drop-down menu at the top of the user interface. At the 5-minute timescale or lower, the user is given the choice to view a particular date instead of the whole timespan, since those levels are very large and the assumption is that only a portion will be watched. Once a date is selected, it will stay selected while the user navigates down to lower levels, making it easier to 'drill down' on interesting content. Also, the user can stay on one level and easily select a different day on that same level to view, which helps for comparing dynamics across different days. When a user hovers to view a thumbnail, the image also displays the date and time (or timespan) represented by that frame. This information is valuable for following the thread of an event from upper to lower levels to gain greater detail and pinpoint the timing of that event. Quick access to date and time information also allows the user to make use of their own knowledge of specific past events at that location or patterns of life relevant to that scene.

5 Results

To demonstrate our approach, we scraped or downloaded 10 datasets captured by outdoor webcams, with lengths ranging from 30 days to 16 years. Details for these datasets are included in the supplemental material, and a quick visual reference is included in Fig. 6. The pyramid videos and spectrograms revealed interesting dynamics and structure at a range of timescales. Below are some general observations, as well as specific findings for several datasets.

5.1 Cycles and Visualization of Periodicity

Events that happen repeatedly stand out with clarity in our pyramid videos. The day/night cycling is the obvious example, and this is noticeable in all of our datasets. However, we found many other examples of cyclic activity that appeared in the videos as distinctive repetitive patterns. Tidal patterns were apparent in the Buxton oceanside dataset as well as the Geiranger ferry dock dataset. Geiranger shows boats rising and falling next to a dock. In the timescales below 1-day, the boats move up and down as the video progresses, and in the timescales longer than 1-day the movement becomes averaged and the video shows a ghostly blur encompassing all of the vertical positions of the boat over time. The cyclic nature of seasons becomes obvious in the datasets with multiple years, such as Hiuchi, Kutcharoko, and Smoky, where

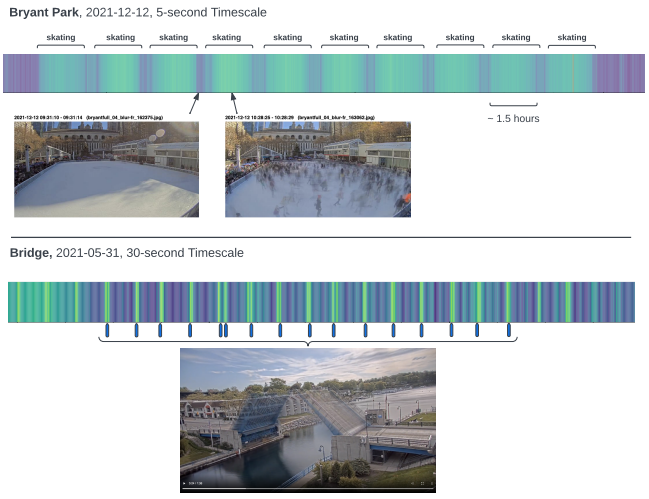


Fig. 7. Examples of periodic human activity showing in the spectrogram.

we can view the seasons changing fast enough that we understand the similarities of the cycle from year to year. The switching between white snowy winter and green leafy summer becomes a visual rhythmic pulse at the higher pyramid levels, just like the day/night changes at the lower pyramid levels.

On a smaller scale, the natural cycles of plant growth are nicely visualized in the Mad River dataset. This scene includes a deciduous tree in the foreground, as well as bushes and other trees on the edge of a river. The tree in the foreground loses its leaves and grows them again, and the bushes and other plants can be viewed getting larger in the summer and smaller in the winter. Another very interesting discovery is how the branches of the foreground tree droop at night and perk up during the day, which becomes more noticeable because it happens repetitively. This dataset seems to have its camera recording with infrared at night, which fortunately makes the tree always visible.

In addition to the pyramid videos themselves, the Video Spectrogram also seems to be especially good at visualizing cyclic events. The day/night cycle is very obvious at the 12-hour timescale, with (usually) more activity and a lighter color on the spectrogram for the half of the day which is mostly daylight, and (usually) a darker color for the less-active night. These two colors on the spectrogram switch back and forth creating a distinctive pattern at that level of the pyramid for most of our datasets. In the higher levels of the multiple-year datasets, the seasonal pulsing is also clearly visualized with the spectrogram colors.

We also found other examples where the periodicity of human activity showed up clearly in the spectrogram (Fig. 7). In the Bryant Park ice rink dataset, the spectrogram had lighter colors during the times the rink was busy with skaters and darker colors for the times when the ice was cleared. This seemed to happen on a cycle of about an hour and a half, presumably a planned timing. Another example is from the Bridge dataset, where the spectrogram shows a light bar every time the drawbridge goes up which happened frequently on Memorial Day in 2021. Since this was a repeating event over the course of that day, its pattern on the spectrogram was more noticeable than it was on the days where the drawbridge only went up once or twice.

5.2 Multiscale Visualization and Drill-Down Navigation

The video spectrogram tool connects events at different timescales by virtue of using a common timeline. If the viewer sees a short blip of some interesting or anomalous event at a higher timescale, the user allows for pinpointing the general date/time of the anomaly and drilling down to lower levels at that same time in order to see more detail. For example, one might see a truck appear 'out of nowhere' in the 6-day timescale, then quickly drill down to the day it appeared and view the 5-second timescale on that day in order to see which direction the truck

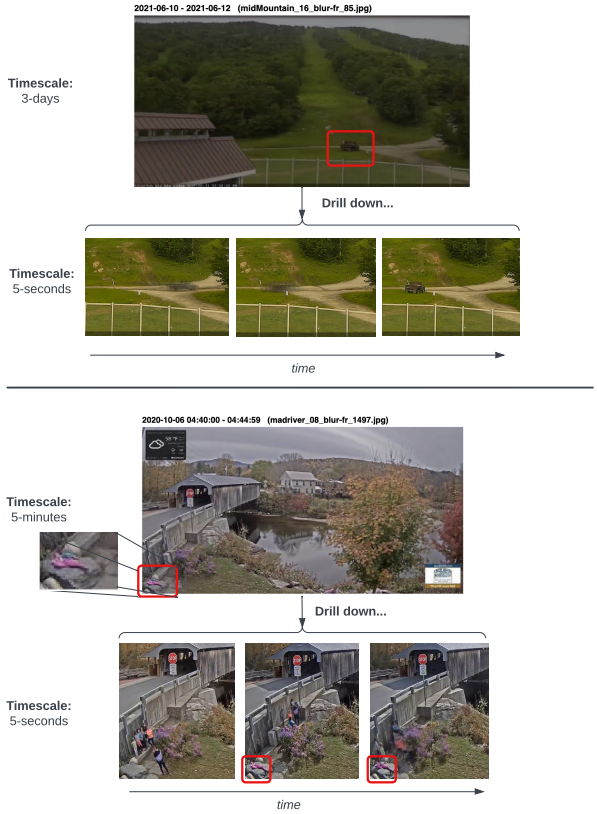


Fig. 8. TOP: In Mid Mountain, a truck abruptly appears and quickly disappears in the 3-day timescale. Drilling down to the 5-second level is necessary to learn that the truck drives in from the left and backs into its parking spot. BOTTOM: In Mad River, a pink jacket left on a rock shows up briefly at the 5-minute timescale. Drilling down reveals a group of people moving around.

drove in from before it parked (Fig. 8, top).

Most real-world events do not fall neatly into one discrete timescale, and this often means that an interesting event can first be discovered from the bird's-eye perspective of a higher pyramid level and then the full extent of the occurrence can be discovered and viewed by drilling down to lower levels. We found an example of this in the Mad River dataset (Fig. 8, bottom), when a bright pink object catches the eye briefly and then disappears during the 5-minute timescale on October 6, 2020, in a corner of the scene. Drilling down to the 5-minute level, the pink object appears to be a pink jacket left on a rock but it still disappears quickly. Drilling down to the 1-minute level we can see fast-moving people and we also see that the pink jacket moves from one rock to another. However, it is not until we drill down to the 5-second timescale that we can make out the group of people and their general movements. The fact that they left a bright pink jacket in one place for about 15 minutes, while they themselves moved faster, left a clue to their presence in that longer 15-minute timescale video.

The multiple timescale visualization also provides a useful and possibly educational demonstration of the role and scope of human activity in a particular scene. In the Bryant Park dataset, the lower level pyramid videos show crowds of people skating. However, the higher level pyramid videos show an eerily empty skating rink, with no humans in sight. At the 4-hour timescale and above, it is mainly the lighting changes, and certain infrastructure changes that stand out (such as a rink-side tent being erected for a while). In the Mid Mountain ski slope dataset, tire tracks and ski tracks in the snow provide a clue to human activity at the lower levels of the pyramid, but we don't see the humans making those tracks unless we drill down. The same thing can be seen with tire tracks and footprints in the sandy beach of the Buxton

dataset. With both snow and sand, the evidence of human activity is melted or eroded away fairly quickly. In contrast, the Rane construction dataset shows a scene where the long term effect of human activity is exactly the point, and in that case it is definitely instructive to drill down and see exactly which lower-timescale activities were responsible for the higher-timescale view of a building being constructed.

5.3 Discovery of Anomalies

The pyramid videos by themselves, as well as the spectrogram tool, can be useful for surfacing anomalous events. Missing data is the most obvious anomaly to find, and sections of missing data show up as all-black frames in the videos and as solid dark colored areas of the spectrogram, as can be seen in Fig. 5. Corrupted data is another kind of anomaly that shows up in the videos and the spectrogram. For instance, in the Buxton coastline dataset the high-level videos have a section that shows a static night scene which is clearly out of place since it lasts longer than a single night both in the pyramid video and on the timeline. We traced the problem back to the original footage, where a static image was looped for a while.

Many anomalies are related to the camera itself, the most common of which is a sudden change in camera angle or placement. The Mid Mountain dataset includes a few months during ski season where the camera zooms or moves closer to the ski slope. The Buxton beach video includes a section where the camera faces out to the ocean instead of along the coastline. Sometimes camera errors can be seen, such as a day in the Hiuchi dataset which included camera footage of an office ceiling when normally the scene is outdoors in the mountains. That event occurred directly after a long period of missing data, so we suspect the camera was being repaired before being reinstalled outdoors. When there is a rainstorm or snowstorm, the video will often show raindrops on the camera lens for a short while at the lower timescales. The most entertaining camera-related anomalies we found occured when birds perch in front of the camera (Buxton) or a spider builds a web on it (Mad River).

In contrast with standard timelapse, the pyramid videos are visually less cluttered and they can be upsampled to adjust the rate of change to be easier to absorb. Anomalous events are more distinctive against this backdrop and are thus fairly easy to spot. For instance, when watching the Buxton dataset video for the 4-hour timescale, we noticed that a railing suddenly appears directly underneath the camera. At that level we could localize it to May 14, 2020, using the Video Spectrogram. We went directly to the 5-minute timescale for May 14, 2020, but the railing was there at the beginning of the day, so we switched to May 13, 2020, and drilled down further. We could see the railing put into place in the real-time original video for May 13, 2020. Even then it was very fast and rather anti-climactic since there was no visual of the person putting it in place (see Fig. 9).

5.4 Long-term Dynamics and Understanding

Our pyramid videos provide a window into the reality of long-term dynamics without sacrificing much verisimilitude. There is some level of blurring that occurs at the upper levels of the pyramid, since it has averaged a lot of small changes over time. We also see the blending of day and night scenes. These factors mean we sacrifice some precision and spatial resolution at the longer timescales; however, we found we were still able to discern larger patterns. For instance, in our Mid Mountain dataset, the gradual pattern of snow melt on a ski slope over the course of days or weeks stands out with clarity at the 6-day timescale. In our Buxton oceanside dataset, at the 1-month timescale, the water's edge can be seen slowly changing its position relative to the beach, slowly rising and retreating much slower than the tides.

Another way these pyramid videos contribute to understanding of long-term dynamics is through the knowledge that anything showing up in a particular timescale must have generally stayed in the same place for a long enough time, related to the timescale. In the 1-hour timescale, a car driving by would not show up but a car parked in one spot for at least an hour would show up, and the duration of its appearance in the video would correspond to how long it stayed parked.

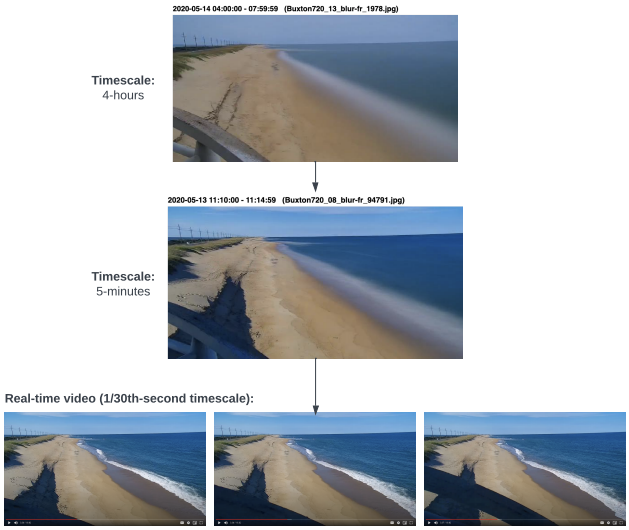


Fig. 9. In the Buxton dataset, at the 4-hour timescale, a railing quickly and obviously appears directly under the camera. Drilling down, we had to go to the original real-time recording to see it being put into place.

5.5 Direct Comparison with Timelapse

For a few datasets, we constructed standard timelapse videos by subsampling at different rates and compiling the resulting frames together into videos for each timescale. At the top levels of the pyramid this resulted in extremely short timelapse videos (less than one second), which were thus not very informative or interesting. However, going down the pyramid levels, once the videos were at least a few seconds long they did provide a good baseline for comparison with our pyramid videos. We found consistent results among all of the datasets which are summarized below.

5.5.1 High Levels: 1-day Timescale and Above

The visual smoothness of our pyramid videos stands in stark contrast with the timelapse videos. At the higher levels especially, each frame of the timelapse is far removed in time from its neighboring frames, increasing the likelihood of major discontinuities in lighting, weather, and other large scene elements. The timelapse videos show the viewer all of these images in rapid succession and the effect is visually chaotic. Only the most obvious changes can get absorbed by the viewer. The rest of the changes are likely to get lost in the noise.

Also, smoothly upsampling our videos to spread out over a longer duration makes them more informative and watchable than the timelapse videos, even after using the video player tools to slow the timelapse videos down to quarter speed. At the 3-day timescale, the timelapse video duration was 4 seconds. Slowing it down to quarter time extended it to 16 seconds. However, our upsampled pyramid video for that timescale was 24 seconds.

5.5.2 Middle Levels: 2-hour to 12-hour Timescales

The timelapse videos from these levels are almost unwatchable because of the strobe effect caused by rapid switching between day and night as the video progresses. This problem would likely be fixed by the complete removal of night-time frames. We did not test that idea, but we believe that even with night frames removed, the timelapse videos from these levels would still suffer from similar faults as they do in the higher and lower levels. Also, for a fair comparison we would also need to remove the night frames from our pyramid videos, and this would likely improve the watchability of those videos as well. We have currently bypassed the strobe effect problem in our pyramid videos by upsampling them so they take longer to watch while the pulsing from day to night happens at a gentler cadence. If we removed night frames,

we would not need to upsample so much and could watch shorter videos. However, for the majority of our datasets there is interesting visual activity during the night hours which we would not want to arbitrarily excise from the video for the sake of watchability or efficiency.

5.5.3 Low Levels: 1-hour Timescale and Below

The timelapse videos are more watchable and informative at lower levels than they are at the higher levels. When comparing a timelapse video with a non-upsampled pyramid video (i.e., of the same length), the pyramid video provides only a slightly better viewing experience because of its smoothness. Upsampling our pyramid video definitely improves its viewing quality.

The most interesting comparison occurs at the very lowest levels (1-minute timescale and below), where we can watch videos for one day at a time and see fast-moving activity such as people skiing and cars driving. In the timelapse videos, fast moving people and cars end up aliased, meaning they show up as a completely solid object and disappear quickly, without a clear trajectory. By contrast, in the pyramid videos, fast moving people and cars will show up as a line of ghostly versions of themselves, along their trajectory. They will only solidify if they stay in one place for long enough. This provides the viewer with more information than the timelapse videos provide. As an example, in the Rane construction site dataset car traffic can be seen on the road in front of the building site at the 15-second timescale. In the timelapse video, we see cars at night and during the day and we can get a general sense that there is less traffic at night. However, in the pyramid video the day/night traffic difference is clearer. We can barely register a blur for night-time traffic, and there is clearly more traffic during the day. It is blurry for the most part, except for when cars stop at the stoplight at regular intervals, at which point they 'solidify' into a clear line of cars. There is a rhythm to this blurred/not-blurred traffic, presumably corresponding with the traffic light schedule. Also, when the cars are stopped we can see that there are almost always more cars in the right lane (possibly getting ready to turn right) This is more information than we would ever glean from the timelapse videos and provides a sound argument for the very basis of our video temporal pyramid.

6 Discussion, Limitations, and Future Work

This work proposes a novel way to visualize the passage of time and explore videos that are too large to be practically explored using traditional tools. We also specifically address phenomena that occur at much longer timescales than most existing methods; these phenomena are present and interesting in our application due to the extreme duration of our videos. Our method compares favorably to naive timelapse videos. Even as an imperfect visualization tool, timelapse videos have been put to good use in a diverse range of applications, such as construction site monitoring [40,47], environmental monitoring [16,23,35], art [14], education [27,42], ecological awareness [6,20], and more. An improvement to existing timelapse techniques could benefit all of these existing applications and possibly lead to interesting new applications. It might also help facilitate a shift in perspective towards long-term thinking. Humanity's short-term or 'real time' bias can make it difficult to tackle long-timescale issues like climate change or urban sprawl. If we don't *see* it happening, we don't care about it as much. Visualization tools can help us *see* it [26,27].

6.1 Limitations

Our method has some important limitations. One is the requirement that the camera viewpoint be fixed. This ensures that changes in the video are due to the scene, rather than camera motion; however, if camera angle changes are infrequent then the spectrogram is minimally affected, and in fact our method is useful for discovering unusual camera events such as a change in viewpoint as discussed in Sect. 5.3.

Another limitation is that the top two or three levels of the pyramid are usually not very informative because there are not enough frames available for any changes to register when stitching the frames together. In a pyramid built from one year's worth of data, the 90-day timescale will only have 4 representative frames. However, in a pyramid built from 8 year's worth of data, the 90-day timescale will include 32 representative frames. We can discern changes over 32 frames much more easily than over 4 frames.

We also noticed that as the pyramid levels progress higher, the edges of all scene elements tend to become slightly less sharp with each new level. This is probably caused by very small camera movements that register as brief 'whole scene changes' with an effect in the pyramid that compounds as levels are built recursively. This effect could possibly be mitigated by the addition of a video stabilization preprocessing step. Simple feature matching-based image alignment techniques [38] could be used to align the frames to minimize movement due to camera shakiness. A similar feature matching technique could be used to manage camera viewpoint changes as well. We found that the noise present in our datasets was small enough that these techniques were not necessary, but they could be used to boost visual quality if desired. They could also help our method generalize to more datasets, such as those with automated and periodic changes in viewpoint, or fixed-viewpoint cameras that exhibit noticeable motion due to wind.

6.2 Future Work

Our temporal pyramid computes a very simple, low-level measure of intensity change from one frame to the next. In the spirit of Viz-a-Vis [32], we intend to explore more sophisticated types of analyses that can be aggregated into heatmaps to show more high-level and/or task-specific measures of activity. For example, optical flow could be used to measure motion rather than per-frame intensity change at multiple timescales (similar to [45]). In scenes with specific object categories of interest (e.g., people, cars etc.), object detection or crowd counting techniques could extract more meaningful trends which could then be visualized in a similar time-frequency spectrogram.

Although our method was not designed with the intent of video anomaly detection, it could provide the basis for some new techniques in that area. One of the many existing video anomaly detection methods [28] could possibly be applied to upper level pyramid videos in order to quickly and automatically surface unusual events at those timescales, which might yield insight upon drill-down to lower levels. For instance, detecting the origin of an unattended bag, after the fact, would be likely made easier with the aid of a temporal pyramid.

Another direction for future work is to explore different types of visualizations for our spectrogram, other than a heatmap. For example, a circular or radial representation might be useful for visualizing periodic events. We would also like to find ways to more easily compare different days with each other (or different years, months, etc.), even if the days chosen for comparison are far removed from each other in time.

7 Conclusion

In this paper, we presented the Video Temporal Pyramid – a multi-scale lens through which to view the passage of time via a process that distills activity happening at different timescales in long fixed-camera video streams. We also presented the Video Spectrogram, a time-frequency visualization to facilitate exploration and discovery in our pyramids. The pyramid videos present a novel alternative to standard timelapse techniques, providing a smooth viewing experience that allows for the absorption of more information about how a scene changes over time. And the spectrogram visualization is the first example of what we believe is a more general and potentially useful class of time-frequency representations for video visualization.

Acknowledgments

This work was supported in part by NASA Award NNX15AJ98H under the Washington NASA Space Grant Consortium, and in part by the National Science Foundation under Grant No. 2105372. The Washington NASA Space Grant Consortium is funded by the NASA Office of Stem Engagement. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA or the NSF.

The authors wish to thank Ann Tseng and Richie Mohan for their early contributions.

REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Human-computer interaction series. Springer, 2011.

[2] M. Ali, M. W. Jones, X. Xie, and M. Williams. Timecluster: Dimension reduction applied to temporal data for visual analytics. *Vis. Comput.*, p. 1013–1026, June 2019.

[3] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. Video tapestries with continuous temporal zoom. *ACM Trans. Gr.*, 29(3), Aug. 2010.

[4] E. P. Bennett and L. McMillan. Computational time-lapse video. *ACM Trans. Gr.*, 2007.

[5] R. Borgo, M. Chen, B. Daubney, E. Grundy, G. Heidemann, B. Höferlin, M. Höferlin, H. Jänicke, D. Weiskopf, and X. Xie. A Survey on Video-based Graphics and Video Visualization. In N. John and B. Wyvill, eds., *Eurographics 2011 - State of the Art Reports*. The Eurographics Association, 2011. doi: 10.2312/EG2011/stars/001-023

[6] E. M. B. Buckley, C. R. Allen, M. L. Forsberg, M. C. Farrell, and A. J. Caven. Capturing change: the duality of time-lapse imagery to acquire data and depict ecological dynamics. *Ecology and Society*, 22:1–12, 2017.

[7] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[8] P. J. Burt. Fast filter transform for image processing. *Computer Graphics and Image Processing*, 1981.

[9] E. Cakmak, U. Schlegel, D. Jäckle, D. Keim, and T. Schreck. Multiscale snapshots: Visual analysis of temporal summaries in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):517–527, 2021.

[10] L. Cohen. *Time-frequency analysis*, vol. 778. Prentice-Hall, Inc., USA, 1995.

[11] A. Finkelstein, C. E. Jacobs, and D. H. Salesin. Multiresolution video. *ACM Trans. Gr.*, 1996.

[12] Google. *Exploring Timelapse in Google Earth*.

[13] C. Gutwin, M. van der Kamp, M. S. Uddin, K. Stanley, I. Stavness, and S. Vail. Improving early navigation in time-lapse video with spread-frame loading. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[14] S. Hansen. Video installation as a creative means of representing temporality in visual data. *Qualitative Research in Psychology*, 15(2-3):292–297, 2018. doi: 10.1080/14780887.2018.1430011

[15] M. Harrower. Visualizing change: Using cartographic animation to explore remotely-sensed data. *Cartographic Perspectives*, (39):30–42, Jun. 2001. doi: 10.14714/CP39.637

[16] B. W. Hartill, S. M. Taylor, K. Keller, and M. S. Weltersbach. Digital camera monitoring of recreational fishing effort: Applications and challenges. *Fish and Fisheries*, 21(1):204–215, 2020. doi: 10.1111/faf.12413

[17] M. Höferlin, K. Kurzhals, B. Höferlin, G. Heidemann, and D. Weiskopf. Evaluation of fast-forward video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2095–2103, 2012. doi: 10.1109/TVCG.2012.222

[18] D. Jackson, J. Nicholson, G. Stoeckigt, R. Wrobel, A. Thieme, and P. Olivier. Panopticon: A parallel video overview system. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13. Association for Computing Machinery, 2013.

[19] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen. Real-time hyperlapse creation via optimal frame selection. *ACM Trans. Gr.*, 2015.

[20] M. Kotack. 4 environmental time-lapse videos you have to see to believe, Oct 2014.

[21] M. Kratochvíl, P. Veselý, F. Mejzlík, and J. Lokoč. SOM-hunter: Video browsing with relevance-to-SOM feedback loop. In *MultiMedia Modeling*. Springer International Publishing, 2020.

[22] Z. Lan, X. Li, and A. G. Hauptmann. Temporal extension of scale pyramid and spatial pyramid matching for action recognition, 2014. doi: 10.48550/ARXIV.1408.7071

[23] J. Liu, R. Chen, C. Han, and W. Qing. Two-year comparative study of snow cover dynamics and its impact factors on glacier surface. *Environmental Earth Sciences*, 75(3):1–11, 2016.

[24] M.-J. Lobo, C. Appert, and E. Pietriga. Animation plans for before-and-after satellite images. *IEEE Transactions on Visualization and Computer Graphics*, 25(2):1347–1360, 2019. doi: 10.1109/TVCG.2018.2796557

[25] R. Martin-Brualla, D. Gallup, and S. M. Seitz. Time-lapse mining from internet photos. *ACM Trans. Gr.*, 2015.

[26] B. Monea, A. Stornaiuolo, and E. P. Catena. Reframing temporality in participatory visual research with timelapse video. *Qualitative Research*, 2021. doi: 10.1177/14687941211019524

[27] K. W. Nakamura, A. Fujiwara, H. H. Kobayashi, and K. Saito. Multi-timescale education program for temporal expansion in ecocentric education: Using fixed-point time-lapse images for phenology observation. *Education Sciences*, 9(3), 2019. doi: 10.3390/educsci9030190

[28] R. Nayak, U. C. Pati, and S. K. Das. A comprehensive review on deep learning-based methods for video anomaly detection. *Image and Vision Computing*, 106:104078, 2021. doi: 10.1016/j.imavis.2020.104078

[29] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1971–1984, 2008. doi: 10.1109/TPAMI.2008.29

[30] A. Rav-acha, Y. Pritch, D. Lischinski, and S. Peleg. Evolving time fronts: Spatio-temporal video warping. Technical report, 2005.

[31] M. Rochan, L. Ye, and Y. Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 347–363, 2018.

[32] M. Romero, J. Summet, J. Stasko, and G. Abowd. Viz-a-vis: Toward visualizing video through computer vision. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1261–1268, 2008.

[33] L. Rossetto, R. Gasser, J. Lokoč, W. Bailer, K. Schoeffmann, B. Muenzer, T. Souček, P. A. Nguyen, P. Bolettieri, A. Leibetseder, and S. Vrochidis. Interactive video retrieval in the age of deep learning – detailed evaluation of vbs 2019. *IEEE Transactions on Multimedia*, 23:243–256, 2021.

[34] M. Rubinstein, C. Liu, P. Sand, F. Durand, and W. T. Freeman. Motion denoising with application to time-lapse photography. In *CVPR*, 2011.

[35] B. Seyednasrollah, A. M. Young, K. Hufkens, T. Milliman, M. A. Friedl, S. Frolking, and A. D. Richardson. *Scientific Data*, 6(1). doi: 10.1038/s41597-019-0229-9

[36] L. Shao, X. Zhen, D. Tao, and X. Li. Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics*, 44(6):817–827, 2014. doi: 10.1109/TCYB.2013.2273174

[37] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996. doi: 10.1109/VL.1996.545307

[38] R. Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007.

[39] Temponaut. *TEMPONAUT Top 10 Timelapses 2021*.

[40] A. Tibaut and D. Zazula. Sustainable management of construction site big visual data. *Sustainability Science*, 13(5):1311–1322, 09 2018.

[41] United States Geological Survey. Repeat Photography Project, 2016.

[42] M. Vollmer and K.-P. Möllmann. Slow speed—fast motion: time-lapse recordings in physics education. *Physics Education*, 53(3):035019, mar 2018. doi: 10.1088/1361-6552/aaa954

[43] P. Wang, Y. Cao, C. Shen, L. Liu, and H. T. Shen. Temporal pyramid pooling-based convolutional neural network for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2613–2622, 2017. doi: 10.1109/TCSVT.2016.2576761

[44] Y. Wang, M. Long, J. Wang, and P. S. Yu. Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[45] S. Wehrwein, K. Bala, and N. Snavely. Scene summarization via motion normalization. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2495–2501, 2021. doi: 10.1109/TVCG.2020.2993195

[46] Wikipedia contributors. Time-lapse photography — Wikipedia, the free encyclopedia, 2022.

[47] J. Yang, M.-W. Park, P. A. Vela, and M. Golparvar-Fard. Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future. *Advanced Engineering Informatics*, 29(2):211–224, 2015. Infrastructure Computer Vision. doi: 10.1016/j.aei.2015.01.011

[48] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.

[49] X. Zhang, J.-Y. Lee, K. Sunkavalli, and Z. Wang. Photometric stabilization for fast-forward videos. *Computer Graphics Forum*, 36(7):105–113, 2017. doi: 10.1111/cgf.13276

[50] Z. Zheng, G. An, D. Wu, and Q. Ruan. Spatial-temporal pyramid based convolutional neural network for action recognition. *Neurocomputing*, 358:446–455, 2019. doi: 10.1016/j.neucom.2019.05.058

[51] F. Zhou, S. B. Kang, and M. Cohen. Time-mapping using space-time saliency. In *CVPR*, 2014.